

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problems Mailbox.**

**DATA TRANSFER METHOD FOR MASTER-SLAVE SYSTEM**

Patent Number: JP1265349  
Publication date: 1989-10-23  
Inventor(s): NAKATANI MITSUYOSHI  
Applicant(s): FUJI ELECTRIC CO LTD; others: 01  
Requested Patent: ☐ JP1265349  
Application Number: JP19870325974 19871223  
Priority Number(s):  
IPC Classification: G06F13/42  
EC Classification:  
Equivalents:

---

**Abstract**

---

**PURPOSE:** To perform the transfer of data at a high speed by securing such a constitution where a slave transmits an answer signal to a master before the transfer of data when the data are transferred to the master from the slave.

**CONSTITUTION:** A master 1 transmits a data transfer request signal 3 as well as an address signal 4 and a slave 2 transmits an answer signal 5 before the output of the data 6. The master 1 receives the signal 5 and stops the output of the address 4 and the request 3. Then the master 1 transmits the next address signal 4 and request 3. While the slave 2 sends the data signal 6 to a bus when a fixed time T1 passed after the output of the signal 5. Then the master 1 fetches the data 6 after the reception of the signal 5 and while the data 6 is fixed on a bus B. In such a way, the data 6 is separated from the signal 5 and this signal 5 is previously informed to the master 1. Thus the master 1 can transmit the next data transfer request.

---

Data supplied from the esp@cenet database - I2

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-194994

(43) 公開日 平成11年(1999) 7月21日

(51) Int. Cl. <sup>6</sup>	識別記号	F I
G06F 13/16	520	G06F 13/16 520 A
13/42	350	13/42 350 Z
15/16	360	15/16 360 R

審査請求 有 請求項の数10 O L (全19頁)

(21) 出願番号 特願平10-428

(22) 出願日 平成10年(1998) 1月5日

(71) 出願人 000168285

甲府日本電気株式会社

山梨県甲府市大津町1088-3

(72) 発明者 神宮寺 聡

山梨県甲府市大津町1088-3 甲府日本電  
気株式会社内

(74) 代理人 弁理士 京本 直樹 (外2名)

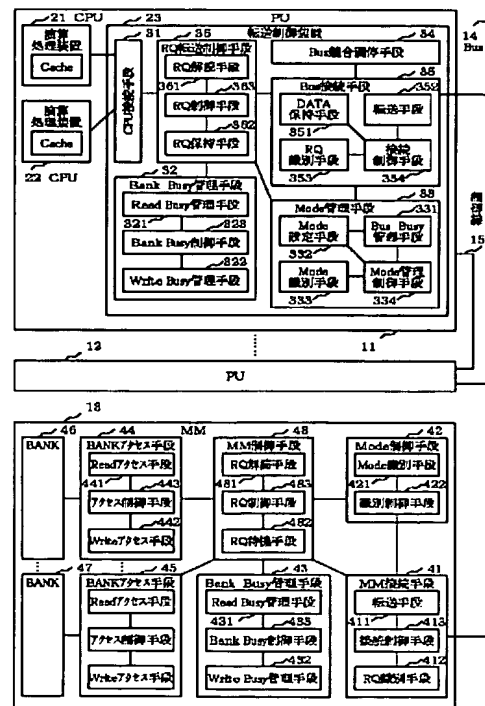
(54) 【発明の名称】 情報処理装置

(57) 【要約】

【課題】 Pipeline方式を用いたシステムバスを持つシステムにおいて、メモリへのWrite動作時のRQとData間のLatencyを可変にする。

【解決手段】 1つ以上の演算処理装置と前記演算処理装置のデータ転送動作を制御する転送制御装置とから構成される処理装置と、1つ以上の前記処理装置から共有して使用され記憶単位であるBankで構成される主記憶部と、前記処理装置間を接続する制御線と、前記処理装置間と前記主記憶部とを接続するパイプライン方式のBusと、前記演算処理装置からデータの転送命令を受け付けたとき前記Busの使用状況から前記転送命令と転送するデータとの時間間隔であるLatencyを設定するモード管理手段を備えた前記転送制御装置とで構成する。

【効果】 RQとData間のLatencyを可変にすることによりシステムバスの利用効率をあげ、システム性能を向上させる。



## 【特許請求の範囲】

【請求項 1】 演算処理装置と転送制御装置とを構成する処理装置からバスを經由して他の処理装置または主記憶部とデータ転送を行う情報処理装置において、1 つ以上の演算処理装置と前記演算処理装置からデータを転送する時間間隔を可変に制御する転送制御装置とから構成される処理装置と、1 つ以上の前記処理装置から共有して使用され記憶単位である Bank で構成される主記憶部と、前記処理装置間を接続する制御線と、前記処理装置間と前記主記憶部とを接続するパイプライン方式の Bus と、を備えることを特徴とする情報処理装置。

【請求項 2】 前記制御線は、前記処理装置が前記 Bus を使用する為の競合調停を行う際に、自処理装置からの前記転送命令を他の全処理装置に送る為に用いられることを特徴とする請求項 1 記載の情報処理装置。

【請求項 3】 前記転送制御装置は、前記演算処理装置からデータの転送命令を受け付けデータ転送を行う CPU 接続手段と、前記主記憶部の Bank のアクセス状況を管理し前記 Bank へアクセスするデータの転送命令の発行時にアクセスが可能か不可能かの判断を行う Bank Busy 管理手段と、前記演算処理装置からデータの転送命令を受け付けたとき前記 Bus の使用状況から前記転送命令と転送するデータとの時間間隔である Mode を設定する Mode 管理手段と、前記 Bus に接続される前記処理装置からの前記制御線を介して Bus に対する前記転送命令の要求の調停を行い前記 Bus の使用権をいずれかの処理装置に与える Bus 競合調停手段と、前記 Bus とインタフェースを持ち前記演算処理装置からのデータの転送を行う Bus 接続手段と、前記 CPU 接続手段と前記 Bank Busy 管理手段と前記 Mode 管理手段と前記 Bus 競合調停手段と前記 Bus 接続手段とを制御する RQ 転送制御手段と、を備えることを特徴とする請求項 1 記載の情報処理装置。

【請求項 4】 前記 Mode 管理手段は、前記 Bus の Data cycle の使用状況を示す Bus 全体の Timing レジスタにより管理する Bus Busy 管理手段と、前記演算処理装置からのデータの転送命令が Write 命令のとき前記 Bus 全体の Timing レジスタの値に対応しデータを前記 Bus に転送する Timing を規定する Mode を設定する Mode 設定手段と、前記 Bus 上に発行された転送命令の前記 Mode を識別し所属する前記処理装置に対し Cache 間または Cache と主記憶部間のデータ転送を行う Timing を判断する Mode 識別手段と、前記 Bus Busy 管理手段と前記 Mode 設定手段と前記 Mode 識別手段とを制御する Mode 管理制御手段と、を備えることを特徴とする請求項 2 記載の情報処理装置。

【請求項 5】 前記主記憶部は、前記 Bank アクセス手段からの制御によりデータを記憶する Bank と、前記 Bus とインタフェースを持ち Bus とのデータ転送を行う MM 接続手段と、前記 Bus からデータの転送命令に設定されている前記 Mode によりデータが送られてくる Timing を識別する Mode 制御手段と、前記処理装置内にある前記 Bank Bus

y 管理手段と同一の機能を持つ Bank Busy 管理手段と、前記 Bank と対をなしデータの転送命令のアクセスを制御する Bank アクセス手段と、前記 MM 接続手段と前記 Mode 制御手段と前記 Bank Busy 管理手段と前記 Bank アクセス手段の制御とデータの転送命令を受け取ったときにこの転送命令のデータを受け取るまで待ち合わせる MM 制御手段と、を備えることを特徴とする請求項 1 記載の情報処理装置。

【請求項 6】 前記 Mode 制御手段は、前記 Bus からの転送命令に設定されている Mode により Data が送られてくる Timing を識別する Mode 識別手段と、前記 Mode 識別手段を制御する識別制御手段と、を備えることを特徴とする請求項 5 記載の情報処理装置。

【請求項 7】 前記転送制御装置は、前記 CPU 接続手段と、前記 Bank Busy 管理手段と、前記 Bus 競合調停手段と、前記 Bus 接続手段と、前記 Bus の使用状況を管理する Bus Busy 管理手段とこの Bus Busy 管理手段を制御する Bus 管理制御手段とから構成される Bus 管理手段と、前記 CPU 接続手段と前記 Bank Busy 管理手段と前記 Bus 競合調停手段と前記 Bus 接続手段と前記 Bus 管理手段とを制御する RQ 転送制御手段と、を備えることを特徴とする請求項 1 記載の情報処理装置。

【請求項 8】 前記 Bus 管理手段は、前記 Bus Busy 管理手段内の Bus 全体の Timing レジスタに加えて有効 Timing レジスタを備えることを特徴とする請求項 4 または 7 記載の情報処理装置。

【請求項 9】 前記有効 Timing レジスタは、Bus 全体の Timing レジスタに登録された Timing が所属する処理装置から前記 Bus に有効なデータの転送命令を出力したことを示し、前記 Bus 全体の Timing レジスタにセットされたとき同じ Timing でセットされることを特徴とする請求項 8 記載の情報処理装置。

【請求項 10】 前記主記憶部は、前記 Mode 制御手段と前記 Bus 管理手段とを置換した構成を備えることを特徴とする請求項 1、5 または 8 記載の情報処理装置。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】本発明は情報処理装置に関し、特に複数の演算処理装置から Pipeline 方式を用いたシステムバスを介してメモリへの Write 動作時にデータ転送時の時間間隔を可変にしシステムバスの使用効率の向上を計る情報処理装置に関する。

## 【0002】

【従来の技術】従来の情報処理装置は、図 16 に従来技術の情報処理装置の構成図を示す。従来技術の情報処理装置は、全て同型の処理装置(以下 PU) 11、12 および図示されていない複数の PU と、複数の記憶単位(以下 Bank) で構成され該 PU に共有される主記憶部(以下 MM) 13 と、これらを接続するシステムバス(以下 Bus) 14 と、該 PU を相互に接続する制御線 15 で構成する。Bus

1 4 は、1つのPUとBus間インタフェースの転送性能と、BusとMM間インタフェースの転送性能が等しく、且つ、MM内のBankインタフェースの転送性能がメモリ素子の動作時間に依存し、且つ、Bus上のリクエスト(以下 RQ)の動作指示を意味するfieldとメモリアクセス時に参照するアドレスを意味するfieldを転送するAddress Cycle、Dataを転送するData Cycleの2 stateで1 Bus Cycleを構成している。該Busの動作はPipeline方式である。Busにおいて、RQから該RQに対応するData(以下 Response Data)は一定時間後に転送されてくる。以後、RQを発行してからデータを転送するまでの時間間隔をLatencyと呼ぶ。また、RQの送出とDataの送出の間、Busは開放された状態となり、この間に別のRQを実行する事が可能である。制御線 1 5 は、各PUがBusを使用する為の競合調停を行う際に、自PUからのRQの種別を他の全てのPUに送る為、及びSystem内のCache coherenceを保つ目的で、Busに発行されたRQに対して自PU内の演算処理装置内にあるCacheのStatusに関する情報を他の全てのPUに送る為に用いられる。Cache Coherency protocolはsnoop方式を用いる。転送制御装置 2 3 は、CPUとインタフェースを持ち該CPUと転送制御装置間のデータ転送を行うCPU接続手段 3 1 と、MMのBankへのアクセス状況を管理し、BankへアクセスするRQ発行時にアクセスが可能か不可能かの判断を行う為に参照されるBank Busy管理手段 3 2 と、Busに接続される各PUからの前記制御線 1 5 を介してBusに対するRQ発行の要求の調停を行い、Busの使用権をいずれかのPUに与えるBus競合調停手段 3 4 と、Busとインタフェースを持ち該Busと前記転送制御装置 2 3 間のデータ転送を行うBus接続手段 3 5 と、CPU接続手段 3 1 とBank Busy管理手段 3 2 とBus競合調停手段 3 4 とBus接続手段 3 5 を制御するRQ転送制御手段 3 6 とで構成する。図 8 は、メモリへのRead動作時のタイムチャートである。図 8 のCycleのA、DはBusのAddress Cycle、Data Cycleを、RQはRead RQを、Dataは該RQのResponse Dataを、LatencyはBusのLatencyをそれぞれ示している。時刻 0 においてRead RQがPU内で処理され、時刻 2 において前記RQはBusに発行され、時刻 4 において前記RQはMM内で処理をされ、時刻 6 から時刻 1 0 間において前記RQはBankアクセスを行い、時刻 1 1 においてMMは前記RQのデータを読みだし、時刻 1 3 において前記データはMM内で転送され、時刻 1 5 において前記データはBusに転送され、時刻 1 7 において前記データはPU内で転送される。RQが発行されてからデータが転送されるまでのBus上のLatencyは、1 2 時刻である。図 1 7 は、メモリへのWrite動作時のタイムチャートである。図のCycleのA、DはBusのAddress Cycle、Data Cycleを、RQはRead RQを、Dataは該RQのResponse Dataを、LatencyはBusのLatencyをそれぞれ示している。時刻 0 においてWrite RQがPU内で処理され、時刻 2 において前記RQはBus発行され、時刻 4 から時刻 1 2 において前記RQはMM内でデータ

の待ち合わせのために持ち回られ、時刻 1 4 から時刻 1 8 においてMMは前記RQのBankアクセスを行い、時刻 1 3 においてライトデータはPU内で転送され、時刻 1 5 において前記データはBusに転送され、時刻 1 7 において前記データはMM内で転送され、時刻 1 9 において前記データはBankに書き込まれる。RQが発行されてからデータが転送されるまでのBus上のLatencyは、Read動作時と同様に 1 2 時刻である。図 1 8 は、あるCPUのメモリへのReadアドレスが他のCPUのCacheにDirtyでHitした場合のCache間データ転送の動作(以下 C Write)時のタイムチャートを示す。図のCycleのA、DはBusのAddress Cycle、Data Cycleを、RQはRead RQを、Dataは該RQのResponse Dataを、LatencyはBusのLatencyをそれぞれ示している。時刻 0 から時刻 1 7 は、図 9 の動作と同様である。ここで、前記Read RQで要求しているアドレスがBusのSnoopの結果、他のPU内のCPUのCacheにDirtyでHitしDirtyでHitしたCPUからRead RQの発行元のCPUへのCache間のデータ転送のRQ(C Write RQ)が生じる。時刻 1 8 においてC Write RQがDirtyでHitしたCPUを持つPU内で処理され、時刻 2 0 において前記RQはBus発行され、時刻 2 2 においてRead RQ発行元のPU内で処理され、時刻 3 1 においてCache間の転送データはC Write RQの発行PU内で転送され、時刻 3 3 において前記データはBusに転送され、時刻 3 5 において前記データはRead RQ発行元のPU内で転送される。RQが発行されてから対応するデータが転送されるまでのBus上のLatencyは、前記Read動作時及び前記Write動作時と同様に 1 2 時刻である。従来の情報処理装置は、図 9、1 0、1 2 のタイムチャートによると、RQの種類に共通して、RQとData間に一定のLatencyがあり、該Latencyがシステム性能に影響を及ぼしていた。また、特開昭 5 7 - 5 0 0 4 4 5 には、システムバスよりのシステムクロックにて特定のコードアドレス及びトランザクション型制御信号の発生及び確認を行うシステムバスの制御により、システムの能力増大と高速化を図る技術が記載されている。また、特開平 0 4 - 3 7 2 0 4 3 には、アドレス信号とデータ信号の出力タイミングをクロックに対して逆位相にすることで最大転送速度を最適化する技術が記載されている。

#### 【 0 0 0 3 】

【発明が解決しようとする課題】 上述した従来の情報処理装置は、第一の問題点は、メモリへのライト動作時、ライト動作を要求するCPUの発行するアドレス情報とライトデータ間はLatencyの時間間隔があり、該Latencyがメモリへのライト動作の性能に潜在的な性能悪化をもたらしている。その理由は、システムバスが、pipeline方式である為である。第二の問題点は、Cache間のデータ転送時、データ転送元のCPUからのアドレス情報と転送データ間はLatencyの時間間隔があり、該LatencyがCache間のデータ転送の性能に潜在的な性能悪化をもたらしている。その理由は、第一の問題点の理由と同様に、シ

システムバスが、pipeline方式である為である。第三の問題点は、Cache間のデータ転送の遅れが、前記転送のDataを受ける側のCPUの動作を一時的に止め複数のCPUを持つシステムの性能低下を引き起こす原因になる。その理由は、第一の問題点の理由と同様に、システムバスが、pipeline方式である為である。

【0004】本発明の目的は、システムバスのLatencyを可変にし前記システムバスの効率的な使用を可能にする情報処理装置を提供することにある。

【0005】

【課題を解決するための手段】第1の発明の情報処理装置は、1つ以上の演算処理装置と前記演算処理装置からデータを転送する時間間隔を可変に制御する転送制御装置とから構成される処理装置と、1つ以上の前記処理装置から共有して使用され記憶単位であるBankで構成される主記憶部と、前記処理装置間を接続する制御線と、前記処理装置間と前記主記憶部とを接続するパイプライン方式のBusと、を備えて構成されている。

【0006】また、第2の発明の情報処理装置は、第1の発明において前記制御線は、前記処理装置が前記Busを使用する為の競合調停を行う際に、自処理装置からの前記転送命令を他の全処理装置に送る為に用いられることにより構成されている。

【0007】さらに、第3の発明の情報処理装置は、第1の発明において前記転送制御装置は、前記演算処理装置からデータの転送命令を受け付けデータ転送を行うCPU接続手段と、前記主記憶部のBankのアクセス状況を管理し前記Bankへアクセスするデータの転送命令の発行時にアクセスが可能か不可能かの判断を行うBank Busy管理手段と、前記演算処理装置からデータの転送命令を受け付けたとき前記Busの使用状況から前記転送命令と転送するデータとの時間間隔であるModeを設定するMode管理手段と、前記Busに接続される前記処理装置からの前記制御線を介してBusに対する前記転送命令の要求の調停を行い前記Busの使用権をいずれかの処理装置に与えるBus競合調停手段と、前記Busとインタフェースを持ち前記演算処理装置からのデータの転送を行うBus接続手段と、前記CPU接続手段と前記Bank Busy管理手段と前記Mode管理手段と前記Bus競合調停手段と前記Bus接続手段とを制御するRQ転送制御手段と、を備えて構成されている。

【0008】さらに、第4の発明の情報処理装置は、第2の発明において前記Mode管理手段は、前記BusのData cycleの使用状況を示すBus全体のTimingレジスタにより管理するBus Busy管理手段と、前記演算処理装置からのデータの転送命令がWrite命令のとき前記Bus全体のTimingレジスタの値に対応しデータを前記Busに転送するTimingを規定するModeを設定するMode設定手段と、前記Bus上に発行された転送命令の前記Modeを識別し所属する前記処理装置に対しCache間またはCacheと主記憶部間のデ

ータ転送を行うTimingを判断するMode識別手段と、前記Bus Busy管理手段と前記Mode設定手段と前記Mode識別手段とを制御するMode管理制御手段と、を備えて構成されている。

【0009】さらに、第5の発明の情報処理装置は、第1の発明において前記主記憶部は、前記Bankアクセス手段からの制御によりデータを記憶するBankと、前記Busとインタフェースを持ちBusとのデータ転送を行うMM接続手段と、前記Busからデータの転送命令に設定されている前記Modeによりデータが送られてくるTimingを識別するMode制御手段と、前記処理装置内にある前記Bank Busy管理手段と同一の機能を持つBank Busy管理手段と、前記Bankと対をなしデータの転送命令のアクセスを制御するBankアクセス手段と、前記MM接続手段と前記Mode制御手段と前記Bank Busy管理手段と前記Bankアクセス手段の制御とデータの転送命令を受け取ったときにこの転送命令のデータを受け取るまで待ち合わせるMM制御手段と、を備えて構成されている。

【0010】さらに、第6の発明の情報処理装置は、第5の発明において前記Mode制御手段は、前記Busからの転送命令に設定されているModeによりDataが送られてくるTimingを識別するMode識別手段と、前記Mode識別手段を制御する識別制御手段と、を備えて構成されている。

【0011】さらに、第7の発明の情報処理装置は、第1の発明において前記転送制御装置は、前記CPU接続手段と、前記Bank Busy管理手段と、前記Bus競合調停手段と、前記Bus接続手段と、前記Busの使用状況を管理するBus Busy管理手段とこのBus Busy管理手段を制御するBus管理制御手段とから構成されるBus管理手段と、前記CPU接続手段と前記Bank Busy管理手段と前記Bus競合調停手段と前記Bus接続手段と前記Bus管理手段とを制御するRQ転送制御手段と、を備えて構成されている。

【0012】さらに、第8の発明の情報処理装置は、第4または7の発明において前記Bus管理手段は、前記Bus Busy管理手段内のBus全体のTimingレジスタに加えて有効Timingレジスタを備えて構成されている。

【0013】さらに、第9の発明の情報処理装置は、第8の発明において前記有効Timingレジスタは、Bus全体のTimingレジスタに登録されたTimingが所属する処理装置から前記Busに有効なデータの転送命令を出力したことを示し、前記Bus全体のTimingレジスタにセットされたとき同じTimingでセットされることにより構成されている。

【0014】さらに、第10の発明の情報処理装置は、第1、5または8の発明において前記主記憶部は、前記Mode制御手段と前記Bus管理手段とを置換した構成を備えて構成されている。

【0015】

【発明の実施の形態】本発明の一つの実施の形態について図面を参照して説明する。図1は本発明の第1の実施

10

20

30

40

50

の形態の構成を示すブロック図である。図 3 は、本発明の第一および第二の実施の形態のRead動作の流れ図である。図 4 は、本発明の第一および第二の実施の形態のWrite動作の流れ図である。図 5 は、本発明の第一および第二の実施の形態のCache間およびCacheとMM間のData転送を行うC Write動作の流れ図である。図 6 は、本発明の第一の実施の形態のRQ/Dataのフォーマットである。図 8 は、本発明の第一、第二の実施の形態および従来技術のRead動作のタイムチャートである。図 9 は、本発明の第一および第二の実施の形態のWrite動作のタイムチャートである。図 1 0 は、本発明の第一および第二の実施の形態のC Write動作のタイムチャートである。図 1 1 は、本発明の第一の実施の形態のRead動作時のBus Busy管理手段におけるレジスタ値の変化の一覧を示す図である。図 1 2 は、本発明の第一の実施の形態のWrite動作時のBus Busy管理手段におけるレジスタ値の変化の一覧を示す図である。図 1 3 は、本発明の第一の実施の形態のWrite動作時のBus Busy管理手段におけるレジスタ値とModeの関係の一覧を示す図である。

【 0 0 1 6 】 本発明の第一の実施の形態は、図 1 を参照すると、全て同型の処理装置(以下PU) 1 1、1 2および図示されていない複数のPUと、複数の記憶単位(以下 Bank)で構成されPU 1 1、1 2に共有される主記憶部(以下 MM) 1 3と、これらを接続するシステムバス(以下 Bus) 1 4と、PU 1 1、1 2を相互に接続する制御線 1 5とから構成される。Bus 1 4は、PU 1 1またはPU 1 2とBus 1 4間インターフェースの転送性能と、Bus 1 4とMM 1 3間インターフェースの転送性能が等しく、且つ、MM 1 3内のBank 4 6、4 7とのインターフェースの転送性能がメモリ素子の動作時間に依存し、且つ、Bus 1 4上のリクエスト(以下 RQ)の動作指示を意味するfieldとメモリアクセス時に参照するアドレスを意味するfieldとを転送するAddress Cycleと、Dataを転送するData Cycleの2 stateで1 Bus Cycleを構成している。Bus 1 4の基本動作はPipeline方式である。MM 1 3に対するRead動作は、RQから一定時間(以下 Latency)後に必ずMM 1 3からResponse Dataが転送されてくる。MM 1 3に対するWrite動作、またはCache間のWrite動作はRQからLatency内に有効なDataが存在しない最速なData Cycle(Latency後のData Cycleが最も遅いData Cycle)でWrite Dataを転送する。いずれのケースでも、RQの送出とDataの送出の間はBus 1 4は開放された状態となり、この間に別のRQを実行することが可能である。制御線 1 5は、PU 1 1、1 2がBus 1 4を使用する為の競合調停を行う際に、自PUからのRQの種別を他の全てのPUに送る為、及びSystem内のCache coherenceを保つ目的で、Bus 1 4に発行されたRQに対して自PU内の演算処理装置内にあるCacheのStatusに関する情報を他の全てのPUに送る為に用いられる。Cache Coherence protocolは、snoop方式を用いる。各PU 1 1、1 2(代表としてPU 1 1で説明)は、同型でCacheを備える演

算処理装置(以下 CPU) 2 1、2 2および図示されていない複数のCPUと、CPU 2 1、2 2とBus 1 4とのデータ転送を制御する転送制御装置 2 3で構成される。転送制御装置 2 3は、CPU 2 1、2 2とインターフェースを持ちCPU 2 1、2 2と転送制御装置間のデータ転送を行うCPU接続手段 3 1と、MM 1 3のBank 4 6、4 7へのアクセス状況を管理し、BankへアクセスするRQ発行時にアクセスが可能か不可能かの判断を行う為に参照されるBank Busy管理手段 3 2と、RQによるBus 1 4のData Cycleの使用状況を管理し、RQとDataのBus Cycleの間隔に関する情報をRQに付加情報(以下 Mode)として設定するMode管理手段 3 3と、Bus 1 4に接続される各PUからの前記制御線 1 5を介してBus 1 4に対するRQ発行の要求の調停を行い、Bus 1 4の使用権をいずれかのPUに与えるBus競合調停手段 3 4と、Bus 1 4とインターフェースを持ち該Busと前記転送制御装置 2 3間のデータ転送を行うBus接続手段 3 5と、CPU接続手段 3 1とBank Busy管理手段 3 2とMode管理手段 3 3とBus競合調停手段 3 4とBus接続手段 3 5を制御するRQ転送制御手段 3 6とから構成されている。

【 0 0 1 7 】 次に転送制御装置 2 3の各手段を説明する。Bank Busy管理手段 3 2は、各PUからのRQをBus競合調停手段 3 4で調停した結果、Busを獲得したPUのRQにより使用されるBankに対してBankの使用状況の登録とアクセスの不可能である時間(以下 Bank Busy)の管理を行う。

【 0 0 1 8 】 Bank Busy管理手段 3 2は、メモリリードRQ(以下 Read RQ)時のBank Busyを管理するRead Busy管理手段 3 2 1と、メモリライトRQ(以下 Write RQ)時のBank Busyを管理するWrite Busy管理手段 3 2 2と、Read Busy管理手段 3 2 1とWrite Busy管理手段 3 2 2を制御するBank Busy制御手段 3 2 3とから構成される。

【 0 0 1 9 】 Mode管理手段 3 3は、Read RQによりResponse Dataが使用するData Cycleと、Write RQおよびCache間およびCacheとMM間のData転送を行うRQ(以下 C Write RQ)により使用されるData Cycleとを各々登録するBus全体のTimingレジスタを備えてBusのData cycleの使用状況を管理するBus Busy管理手段 3 3 1と、Write RQおよびC Write RQを発行するときBus Busy管理手段 3 3 1のData cycleの使用状況を参照しDataをBus 1 4上に発行するTimingを規定するModeを設定するMode設定手段 3 3 2と、Bus 1 4上に発行されたRQのModeを識別し所属するPUに対するCWrite RQのDataがBus 1 4に発行されるTimingを判断するMode識別手段 3 3 3と、Bus Busy管理手段 3 3 1とMode設定手段 3 3 2とMode識別手段 3 3 3とを制御するMode管理制御手段 3 3 4とから構成される。

【 0 0 2 0 】 Bus接続手段 3 5は、Bus競合調停手段 3 4による調停の結果、自PUがBus 1 4を獲得したRQに対応するDataを、Mode管理手段 3 3により設定されたModeの

規定するTimingでBus 1 4に発行されるまで一時的に保持するDATA保持手段 3 5 1と、Bus 1 4とインタフェースを持ちBus 1 4へのデータ転送を行う転送手段 3 5 2と、Bus 1 4からのRQを識別し所属するPUへのRQを検出するRQ識別手段 3 5 3と、DATA保持手段 3 5 1とBus接続手段 3 5 2とRQ識別手段 3 5 3とを制御する接続制御手段 3 5 4とから構成される。

【0 0 2 1】転送制御手段 3 6は、CPUから発行されたRQおよびBusからのRQを解読するRQ解読手段 3 6 1と、RQのアクセスするBankがBusy状態である場合、Busyが解除されるまでの間、または、Bus競合調停手段 3 4による調停の結果によって所属するPUがBusを獲得できなかった場合にRQを一時的に保持し、さらに保持されているRQがある場合、後続のRQを一時的に保持するRQ保持手段 3 6 2と、RQ解読手段 3 6 1とRQ保持手段 3 6 2とを制御するRQ制御手段 3 6 3とから構成される。

【0 0 2 2】次にMM 1 3の構成を図 1を参照して説明する。MM 1 3は、Bus 1 4とインタフェースを持ちBus 1 4とMM 1 3間のデータ転送を行うMM接続手段 4 1と、Bus 1 4からのRQに設定されているModeによりDataの送られて来るTimingを識別するMode制御手段 4 2と、各PU内にある前記のBank Busy管理手段 3 2と同一の機能を持つBank Busy管理手段 4 3と、複数のBank 4 6、4 7と対をなしRead/Writeのアクセスを制御するBankアクセス手段 4 4、4 5および図示されていない複数のBankアクセス手段と、同型で前記Bankアクセス手段 4 4、4 5からの制御によりDataを記憶するBank 4 6、4 7および図示されていない複数のBankと、MM接続手段 4 1とMode制御手段 4 2とBank Busy管理手段 4 3とBankアクセス手段 4 4、4 5を制御し、メモリへのWrite動作のRQを受け取った場合に該RQに対応するDataを受け取るまでRQを待ち合わせるMM制御手段 4 8とから構成される。

【0 0 2 3】次にMM 1 3を構成する各手段を説明する。MM接続手段 4 1は、Bus 1 4からのRQを識別しMM 1 3へのRQを検出するRQ識別手段 4 1 2と、Bus 1 4とインタフェースを持ちBus 1 4へのデータ転送を行う転送手段 4 1 1と、RQ識別手段 4 1 2と転送手段 4 1 1とを制御する接続制御手段 4 1 3とから構成される。Mode制御手段 4 2は、Bus 1 4からのRQに設定されているModeによりDataの送られて来るTimingを識別するMode識別手段 4 2 1と、Mode識別手段 4 2 1を制御する識別制御手段 4 2 2とから構成される。Bank Busy管理手段 4 3は、各PU内にある前記のBank Busy管理手段 3 2と同一の機能を持つRead Busy管理手段 4 3 1とWrite Busy管理手段 4 3 2とBank Busy制御手段 4 3 3とから構成される。各Bankアクセス手段 4 4、4 5(代表としてBankアクセス手段 4 4で説明)は、Bank 4 6からのDataの読みだし動作のアクセスを制御するReadアクセス手段 4 4 1と、Bank 4 6へのDataの書き込み動作のアクセスを制御するWriteアクセス手段 4 4 2と、Readアクセス手段

4 4 1とWriteアクセス手段 4 4 2を制御するアクセス制御手段 4 4 3とから構成される。MM制御手段 4 8は、Bus 1 4からのRQの動作指示を示すTypeからMM 1 3への動作を解読するRQ解読手段 4 8 1と、メモリへのWrite動作のRQを受け取った場合に該RQに対応するDataを受け取るまでRQを待ち合わせるRQ待機手段 4 8 2と、RQ解読手段 4 8 1とRQ待機手段 4 8 2とを制御するRQ制御手段 4 8 3とから構成される。また、システム内のメモリは一つの論理アドレスに定義され、CPUからは、物理アドレスに変換されたアドレスがRQのアドレスとして用いられる。

【0 0 2 4】次に、第一の実施の形態のRead動作とWrite動作とC Write動作を図 1、図 3、図 4、図 5、図 6、図 8、図 9、図 1 0、図 1 1、図 1 2、図 1 3を用いて説明する。CPU 2 1がBank 4 6に対しRead RQを発行した場合のRead動作について説明する。図 6に示すフォーマットは、BusへのRQを示す有効ビット(V)、RQの動作指示(Type)、Write動作時に有効な付加情報(Mode)、アクセスするBank内アドレスとBank番号(Address)の 4 つのfieldで構成される。Read RQは、Read動作指示(命令フォーマットは図示せず)がType fieldに含まれる。Dataは、転送データで構成される。各ステップを以下 Sと記す。

【0 0 2 5】図 3を参照すると、S1でCPU 2 1がRead RQを転送制御装置 2 3に発行する。RQ転送制御装置 3 6では、RQをCPU接続手段 3 1を介しRQ解読手段 3 6 1で受け取る。

【0 0 2 6】S2でRQ解読手段 3 6 1は、RQのTypeからRead RQであることを解読するとRQ制御手段 3 6 3にRead RQに関する制御情報を渡す。

【0 0 2 7】S3でRQ制御手段 3 6 3は、所属するPU内のCPUが以前に発行されたRQがRQ保持手段 3 6 2内に保持されているか確認する。保持されているRQがあれば、S4としてS2で解読されたRQをRQ保持手段 3 6 2の保持されているRQの列の最後尾に保持し、S5としてRQ保持手段 3 6 2にある最も古いRQをS6以降の動作対象として選択する。RQ保持手段 3 6 2で保持されているRQが無ければ、S2で解読されたRQをS6以降の動作対象として選択する。

【0 0 2 8】S6でBank Busy管理手段 3 2は、選択されたRQがRead RQであることを示す情報とアクセスするBank番号を受け取り、Read Busy管理手段 3 2 1を参照し該当するBank(ここではBank 4 6)に対してアクセスが可能かを確認する。アクセス可能であればS7以降の処理を行い、アクセス不可能であればRead RQをRQ保持手段 3 6 2にて一時的に保持する(S4に戻る)。

【0 0 2 9】S7でBus競合調停手段 3 4は、各PUがBusへのRQ発行する際に生じるPU間での競合の調停を行う。この際、Bus競合調停手段 3 4は、所属するPU 1 1を除いた他の全てのPUから調停に必要な情報を制御線 1 5を介して受け取り、同時に他の全てのPUが調停に必要な情



報を他の全てのPUに送る。システム内の全てのPUにおいて、この調停に関する処理は同期している。Bus競合調停手段 3 4 による調停の結果、所属するPUのRead RQがBusに対して発行可能な状態になると、Bank Busy管理手段 3 2 のBank Busyの状態を更新し、S 8 以降の処理を行い、発行不可能な状態のときは該RQをRQ保持手段 3 6 2 にて一時的に保持する(S 4 に戻る)。

【0 0 3 0】S 8 でBus接続手段 3 5 は、RQ転送制御手段 3 6 からRead RQを受け取り転送手段 3 5 2 を用いてBusに対してRead RQを発行する。

【0 0 3 1】S 9 でMode管理手段 3 3 は、Busに対してRead RQを発行してからLatency後に該RQのResponse DataがBus上に存在する、すなわちBusのData Cycleを使用することを意味する情報をBus管理手段 3 3 1 に登録する。Bus Busy管理手段 3 3 1 内のレジスタ値の変化を図 1 1 に示す。Bus全体のTimingレジスタとは、Bus上に有効なDataが存在するTimingの情報を持つ。ここでは、Latencyが7Bus Cycleである場合について説明する。BusにRQが発行される1Bus Cycle前を時刻Tとし、RQが発行された時を時刻T+1とする。時刻T+1では、発行されたRead RQのResponse Dataのタイミング(Latency後)に、Bus上に有効なDataが存在することを示すため、レジスタの最上位に1をセットする。このレジスタを毎Bus Cycleごとに右シフトし、最下位に1が来たタイミングでBusに上記Read RQのResponse DataがBusに存在することを意味する。

【0 0 3 2】S 1 0 でMM接続手段 4 1 は、Bus上のRQを取り込みRQ識別手段 4 1 1 でRQのTypeからRQの識別を行い、MMへのRQである場合はS 1 1 以降の処理を行い、MMへのRQでなければ動作しない。

【0 0 3 3】S 1 1 でMM制御手段 4 8 は、RQ解読手段 4 8 1 にてRQ識別手段 4 1 1 からRQを受け取りさらにRQのTypeを解読し、Read RQである場合はS 1 2 以降の処理を行い、Read RQでなければその他の動作を行う。

【0 0 3 4】S 1 2 でRQ解読手段 4 8 1 は、RQのAddress fieldに含まれるBank番号からアクセスするBankを解読する(ここでは、Bank 4 6 へのアクセスと解読される)。

【0 0 3 5】S 1 3 でBank Busy管理手段 4 3 は、RQ解読手段 4 8 1 からRQがRead RQであることを示す情報をアクセスするBank番号を受け取りRead Busy管理手段 4 3 1 を参照し、該当するBank(ここではBank 4 6)に対してアクセスが可能かを確認する。アクセス可能であればS 1 3 以降の処理を行い、アクセス不可能であれば障害と識別し障害を処理する動作を行う(ここでは障害を処理する動作に関しては言及しない)。また、アクセス可能であるBankに関する情報を受け取ると、Bank Busyの状態を更新する。

【0 0 3 6】S 1 4 でBankアクセス手段 4 4 は、RQ解読手段 4 8 1 からのRead指示をReadアクセス手段 4 4 1 で受け取り、RQのAddress fieldに含まれるBank内アドレ

スで指定されたアドレスからDataを読み出す。

【0 0 3 7】S 1 5 でBankから読み出されたDataは、Bankアクセス手段 4 4、MM制御手段 4 8 を介してMM接続手段 4 1 内の転送手段 4 1 2 からBusへ転送される。このDataは、Read RQがBusに発行されてからLatency後にBusへ転送される。

【0 0 3 8】S 1 6 でRead RQ発行元PU 1 1 は、RQ識別手段 3 5 3 でRQをBus上に発行してからLatency後にBus上にあるDataを該RQのResponse Dataをして受け取る。

10 【0 0 3 9】S 1 7 でRQ転送制御手段 3 6 は、前記Response Dataを受け取りCPU接続手段 3 1 を介してRead RQを発行したCPU 2 1 にDataを返し、一連のRead動作を終了する。

【0 0 4 0】図 8 は、第一の実施の形態におけるRead RQの動作のタイムチャートである。図 8 において、CycleのAはBusのAddress Cycleを示し、CycleのDはBusのAddress CycleのData Cycleを示し、RQはRead RQを示し、Dataは該RQのResponse Dataを示し、LatencyはBusのLatencyを示している。Read RQの動作のタイムチャートにおいて、第一の実施の形態は従来の技術と同じである(従来技術のRead動作時の説明を参照のこと)。

【0 0 4 1】次に、第一の実施の形態のWriteの動作を図 1、図 4、図 6 を用いて説明する。CPU 2 1 がBank 4 6 に対しWrite RQを発行した場合の動作について説明する。

【0 0 4 2】図 4 を参照すると、S 1 でCPU 2 1 がWrite RQ、および該RQに対応するWrite Dataを転送制御装置 2 3 に発行する。RQ転送制御装置 3 6 では、RQとDataをCPU接続手段 3 1 を介しRQ解読手段 3 6 1 で受け取る。

30 【0 0 4 3】S 2 でRQ解読手段 3 6 1 は、RQのTypeからWrite RQである事を解読するとRQ制御手段 3 6 3 にWrite RQに関する制御情報を渡す。S 3 でRQ制御手段 3 6 3 は、所属するPU内のCPUが以前に発行されたRQがRQ保持手段 3 6 2 内に保持されているか確認する。保持されているRQがあれば、S 4 としてS 2 で解読されたRQをRQ保持手段 3 6 2 の保持されているRQの列の最後尾に保持し、S 5 としてRQ保持手段 3 6 2 にある最も古いRQをS 6 以降の動作対象として選択する。RQ保持手段 3 6 2 で保持されているRQが無ければ、S 2 で解読されたRQをS 6 以降の動作対象として選択する。

【0 0 4 4】S 6 でBank Busy管理手段 3 2 は、選択されたRQがWrite RQであることを示す情報とアクセスするBank番号を受け取り、Write Busy管理手段 3 2 1 を参照し該当するBank(ここではBank 4 6)に対してアクセスが可能かを確認する。アクセス可能であればS 7 以降の処理を行い、アクセス不可能であればWrite RQを一時的に保持する(S 4 に戻る)。

【0 0 4 5】S 7 でBus競合調停手段 3 4 は、各PUがBusへのRQ発行する際に生じるPU間での競合の調停を行う。この際、Bus競合調停手段 3 4 は、所属するPU 1 1 を除

いた他の全てのPUから調停に必要な情報を制御線 1 5 を介して受け取り、同時に、他の全てのPUが調停に必要な情報を他の全てのPUに送る。システム内の全てのPUにおいて、この調停に関する処理は同期している。Bus競合調停手段 3 4 による調停の結果、所属するPUのWrite RQがBusに対して発行可能な状態になると、BankBusy管理手段 3 2 のBank Busyの状態を更新し、S 8 以降の処理を行い、発行不可能な状態であると該RQをRQ保持手段 3 6 2 にて一時的に保持する(S 4 に戻る)。

【0 0 4 6】S 8 でBus競合調停手段 3 4 による調停の結果、所属するPUのWrite RQがBusに対して発行可能な状態になると、Mode管理手段 3 3 は、該RQがWrite RQである情報を受け取り、Bus管理手段 3 3 1 でBusのData Cycleの使用状況を確認する。BusはPipeline方式により動作する為、Write RQに対応するWrite DataはWrite RQがBus上に発行されてからLatency後にBus上に転送できるが、Latency以前に使用されていないData Cycleがある場合、Write RQがBusに発行されてから最早の使用されていないData CycleでDataをBusに転送することが可能である。この場合、Write Dataを転送するタイミングすなわち、Data Cycleを使用するタイミングを意味する情報をBus Busy管理手段 3 3 1 に登録し、Mode設定手段 3 3 2 にてWrite RQにData Cycleを使用するタイミングの情報をModeとして設定する。Bus Busy管理手段 3 3 1 内のレジスタ値の変化を図 1 2 に示す(第一の実施の形態におけるRead RQ動作のS 9 参照)。図 1 2 を参照すると、レジスタ値 (T) の状態が現在のBusの使用状態を示している。Data Cycleを使用するタイミングであるData Cycleはレジスタ値の 0 の位置が該当する。例えば、レジスタ値「x x x x x 0」であれば 0 の位置が最右端であるから 1 番目のBus Cycleを使用しBus上にWrite RQが発行されるBusのタイミングであることを示している。同様にレジスタ値「x x x x x 0 1」であれば 0 の位置が 2 番目であるから 2 番目のBus Cycleを使用しBus上にWrite RQが発行されるBusのタイミングであることを示している。このBusのタイミングが図 1 3 に示すModeと対応している。すなわち、Write RQが発行されるとき図 1 2 で示すレジスタ値 (T) の参照したときレジスタ値に対応したModeが決定する。また、Write RQが発行されるときレジスタ値の 0 の位置に 1 がセットされる。前記レジスタ値 (T) 「x x x x x 0」であれば「x x x x x 1」となり、前記「x x x x x 0 1」であれば「x x x x x 1 1」となる。1 Bus Cycleが経過するとレジスタ値は右シフトされた状態がレジスタ値 (T + 1) である。すなわち前記「x x x x x 1」は「0 x x x x x」となり、前記「x x x x x 1 1」は「0 x x x x x 1」となる。最も左のBitにはLatencyではDataが存在しないことを意味する 0 がセットされる。

【0 0 4 7】S 9 でBus接続手段 3 5 は、RQ転送制御手段 3 6 からWrite RQとWrite Dataを、Mode管理手段 3 3 か

らModeを受け取り、Modeによって指定されたData Cycleまでの間、Write DataをData保持手段 3 5 1 で一時的に保持する。

【0 0 4 8】S 1 0 でBus接続手段 3 5 は、転送手段 3 5 2 を用いてBusに対してWrite RQを発行する。

【0 0 4 9】S 1 1 でMM接続手段 4 1 は、Bus上のRQを取り込みRQ識別手段 4 1 1 でRQのTypeからRQの識別を行い、MMへのRQである場合はS 1 2 以降の処理を行い、MMへのRQでない場合は動作しない。

【0 0 5 0】S 1 2 でMM制御手段 4 8 は、RQ解読手段 4 8 1 にてRQ識別手段 4 1 1 からRQを受け取りさらにRQのTypeを解読し、Write RQである場合は、前記Write RQに対応するWrite Dataを受け取るまでRQの有効ビット、Bank内アドレス、Bank番号をRQ待機手段 4 8 2 において一時的に待機させ、S 1 3 以降の処理を行う。Write RQでない場合はその他の動作を行う。

【0 0 5 1】S 1 3 でRQ解読手段 4 8 1 は、RQのAddressに含まれるBank番号からアクセスするBankを解読する(ここでは、Bank 4 6 へのアクセスと解読される)。

【0 0 5 2】S 1 4 でMode制御手段 4 2 では、RQのModeをMode識別手段 4 2 1 で受け取る。Mode識別手段 4 2 1 では、Modeの内容から受け取ったWrite RQに対応するWrite Dataが何Bus cycle後のData CycleにBus上に存在するかを識別する。

【0 0 5 3】S 1 5 でWrite RQ発行元PU 1 1 のBus接続手段 3 5 では、Data保持手段 3 5 1 に保持しているDataを、該RQのModeが指定するData cycleにBusへ転送する。

【0 0 5 4】S 1 6 でMM接続手段 4 1 は、Mode制御手段 4 2 からの指示によりRQ識別手段 4 1 2 にてModeの指示するData CycleにあるDataをBusより受け取る。

【0 0 5 5】S 1 7 でBank Busy管理手段 4 3 は、RQ解読手段 4 8 1 からRQがWrite RQであることを示す情報と、RQ待機手段 4 8 2 からアクセスするBank番号を受け取りWrite Busy管理手段 4 3 1 を参照し、該当するBank(ここではBank 4 6)に対してアクセスが可能かを確認する。アクセス可能であればS 1 3 以降の処理を行い、アクセス不可能であれば障害と識別し障害を処理する動作を行う(ここでは障害を処理する動作に関しては言及しない)。また、アクセス可能であるBankに関する情報を受け取ると、Bank Busyの状態を更新する。

【0 0 5 6】S 1 8 でBankアクセス手段 4 4 は、RQ解読手段 4 8 1 からのWrite指示をWriteアクセス手段 4 4 1 で受け取り、RQ待機手段 4 8 2 からRQのBank内アドレスを受け取り、指定されたアドレスにDataを書き込み、一連のWrite動作を終了する。

【0 0 5 7】図 1 1 は、第一の実施の形態におけるWrite RQの動作のタイムチャートを説明する。図のCycleのA、DはBusのAddress Cycle、Data Cycleを、RQはWrite RQを、Dataは該RQのResponse Dataを、LatencyはBusのL

latencyをそれぞれ示している。図 1 1 では、Write RQが Busに発行されてからWrite DataがBus上に存在するまでの最速のタイミングを実線で表し、ModeによりWrite Dataの存在するData Cycleのタイミングの変化を点線で表す。時刻 0 においてWrite RQがPU内で処理され、時刻 1 においてライトデータはPU内で転送され、時刻 2 において前記RQはBus発行され、時刻 3 において前記データはBusに転送され、時刻 4 において前記RQはMM内で処理され、時刻 5 において前記データはMM内で転送され、時刻 6 から時刻 1 0 においてMMは前記RQのBankアクセスを行い、時刻 1 1 において前記データはBankに書き込まれる。RQが発行されてからデータが転送されるまでのBus上のLatencyは 0 時刻、すなわちLatencyがないことを意味する。図中で点線は、Latencyが 1 Bus Cycle毎遅れた場合を表している。最も遅く前記データがBusに発行されるのは、時刻 1 5 であり、このときのLatencyは、Read動作時のLatencyと等しく 1 2 時刻となる。

【 0 0 5 8 】次に第一の実施の形態で、CPU 2 1 がBank 4 6 に対しRead RQを発行し、該RQが、PU 1 2 内に存在するCache LineにDirtyでHitし、PU 1 2 がPU 1 1 に対しC Write RQを発行した場合のC Write動作について図 4、図 5、図 6、図 9、を用いて説明する。図 5 は第一の実施の形態のC Write RQの動作の流れ図であり、図 3 のRead RQの動作の流れ図と説明および図 4 のWrite RQの動作の流れ図と説明とを引用する。説明には図 1 の名称と符合を使用する。ただし、図 4 のWrite RQの動作の説明を引用する場合、Write RQをC Write RQに差し替える。

【 0 0 5 9 】S 1 でCPU 2 1 がRead RQを転送制御装置 2 3 に発行する。以降の処理は、Read RQの動作の説明S 1 と同様であるので省略する。S 2 では、Read RQの動作の説明S 2 からS 7 と同様であるので省略する。S 3 では、Read RQの動作の説明S 8 と同様であるので省略する。S 4 では、Read RQの動作の説明S 9 と同様であるので省略する。S 5 でRead RQを発行したPU以外の全てのPU(ここでは、PU 1 2)では、システム内のCache Coherencyを保つ為、毎Address CycleにBusをSnoopする。S 6 でPU 1 2 では、SnoopしたRQのAddress fieldと自PU内に存在するCache Lineを比較し、前記Read RQの要求するDataが前記Cache LineにDirtyでHitした場合、S 7 以降の処理を行い、DirtyでHitしない場合、S 8 の処理を行う。S 7 でPU 1 2 は、前記Read RQが自PU内にDirtyでHitし、C Write RQを発行することを他の全てのPU(ここでは、PU 1 1)に対して制御線 1 5 を使って通知する。よって前記Read RQを発行したCPU 2 1 は、前記Read RQによってC Write RQが発行されることを識別する。S 8 でPU 1 2 は、前記Read RQが自PU内にDirtyでHitしなかったことを他の全てのPU(ここでは、PU 1 1)に対して制御線 1 5 を使って通知する。この場合、PU 1 1 は制御線 1 5 の情報による動作はしない。S 9 では、Read RQの動作の説明S 1 0 か

らS 1 4 と同様であるので省略する。S 1 0 では、Read RQの動作の説明S 1 5 からS 1 6 と同様であるので省略する。S 1 1 でRead RQ発行元のCPU 2 1 では、自身の発行したRead RQに対してS 7 でPU 1 2 からC Write RQによりDataが供給されることが通知されているので、MM 1 3 からのResponse Dataを無効にする。S 1 2 でDirtyでHitしたCache Lineを持つCPU(この場合PU 1 2 に所属している)は、Read RQの発行元のPU 1 1 に対して該RQのResponseとしてC Write RQを発行する。S 1 3 は、Write RQの動作の説明S 2 からS 9 と同様であるので省略する。ただし、PU 1 2 に動作である為に符合は参照しない。S 1 4 は、Write RQの動作の説明S 1 0、S 1 1、S 1 4、S 1 5 と同様であるので省略する。ただし、PU 1 2 に動作である為に符合は参照しない。S 1 5 でPU 1 1 は、C Write RQに対してRead RQの動作の説明S 1 6 と同様に動作する。この時、Mode識別手段 3 3 3 により、C Write RQのModeにより指定されたTimingでBusからDataを受け取る。S 1 6 は、Read RQの動作の説明S 1 7 と同様であり、一連のC Write RQ動作を終了する。図 1 0 を参照し、第一の実施の形態におけるC Write RQの動作をタイムチャートで説明する。図 1 0 において、CycleのA、D はBusのAddress Cycle、Data Cycleを、RQはRead RQとC Write RQを、Dataは該RQのResponse DataとC Write RQのDataを、LatencyはBusのLatencyをそれぞれ示している。図 1 0 では、Read RQがBusに発行されたことによりC Write DataがBus上に発行され、該C Write RQによるDataがBus上に最速のタイミングで存在する場合を表す。

【 0 0 6 0 】時刻 0 から時刻 1 8 までは、従来技術のRead動作と同様である(従来技術のC Write RQの動作説明参照のこと)。時刻 1 9 においてライトデータはPU内で転送され、時刻 2 0 において前記RQはBus発行され、時刻 2 1 において前記データはBusに転送され、時刻 2 2 において前記RQはRead RQ発行元のPU内で処理され、時刻 2 3 において前記データはRead RQ発行元のPU内で転送される。C Write RQが発行されてから対応するデータが転送されるまでのBus上のLatencyは 0 時刻、すなわちLatencyがないことを意味する。図中で点線は、Latencyが最も遅れた場合を表している。最も遅く前記データがBusに発行されるのは、時刻 3 3 であり、このときのLatencyは、Read動作時のLatencyと等しく 1 2 時刻となる。

【 0 0 6 1 】次に、本発明の第二の実施の形態について図面を参照して説明する。図 2 は、本発明の第二の実施の形態を示すブロック図である。図 7 は、本発明の第二の実施の形態のRQ/Dataのフォーマットである。図 1 4 は、本発明の第二の実施の形態のRead動作時のBus Busy管理手段におけるレジスタ値の変化の一覧を示す図である。図 1 5 は、本発明の第二の実施の形態のWrite動作時のBus Busy管理手段におけるレジスタ値の変化の一覧を示す図である。

【0062】本発明の第二の実施の形態では、第一の実施の形態における各PU内に含まれるMode管理手段33をBus管理手段49に置き換え、MM13に含まれるMode管理手段42を各PU内に含まれる前記Bus管理手段49に置き換えた構成である。各PU11、12内のBus管理手段49と、MM13内のBus管理手段49は同一の手段である。

【0063】第二の実施の形態でBus管理手段49は、Read RQによりResponse Dataが使用するData Cycleと、Cache間およびCacheとMM間のData転送を行うWrite RQ(以下C Write RQ)により使用されるData Cycleとを各々登録する2種のレジスタと、RQによるBusのData cycleの使用状況を管理するBus Busy管理手段491と、Bus Busy管理手段491を制御するBus管理制御手段492とから構成される。

【0064】次に第二の実施の形態のRead動作とWrite動作とC Write動作を図2、図3、図4、図5、図7、図8、図9、図10、図14、図15を用いて説明する。ただし図3において、第二の実施の形態のRead RQの動作の説明ではS12'が含まれる。図7は、Address/Data Cycleのフォーマットを示し、有効ビット(V)、RQの動作指示(Type)、アクセスするBank内アドレスとBank番号(Address)の3つのfieldで構成される。Read RQは、Read動作指示(命令フォーマットは図示せず)がType fieldに含まれる。Dataは、転送データで構成される。

【0065】CPU21がBank46に対しRead RQを発行した場合のRead動作について図3を用いて説明する。

【0066】S1でCPU21がRead RQを転送制御装置23に発行する。以下第一の実施の形態におけるS1からS12と同様である。

【0067】ただし、S9の動作が異なる。S9ではBus管理手段491内のレジスタ値の変化を図14に示す。図14に示すように、Bus管理手段491内に第一の実施の形態のBus Busy管理手段331内のBus全体のTimingレジスタに加えて有効Timingレジスタを備えている。この有効Timingレジスタは、Bus全体のTimingレジスタに登録されたTimingが所属するPUからBusに有効なRQを出力したことを示し毎BusCycleに右シフトする。この有効Timingレジスタは、時刻T+1において、Bus全体のTimingレジスタにセットされた1が所属するPUに有効であれば同じTimingで1がセットされ、有効でなければ0がセットされる。

【0068】S12'でBus管理手段49は、MM制御手段48からMM13が受け取ったRQがRead RQである情報を受け取り、Bus Busy管理手段491に該RQのResponse DataがBusを使用するData Cycleのタイミングを登録する。Bus Busy管理手段491のレジスタの動作は、S9のBus Busy管理手段491と同様である(ただし、「所属するPU」を「所属するMM」に読み替える)。

【0069】S13からS17では、第一の実施の形態に

におけるRead動作の説明と同様である。以上により、一連のRead動作を終了する。

【0070】図8は、第二の実施の形態におけるRead RQの動作のタイムチャートである。Read RQの動作のタイムチャートにおいて、第二の実施の形態は第一の実施の形態および従来の技術と同じである(第一の実施の形態のRead RQのタイムチャートの説明参照のこと)。

【0071】次に第二の実施の形態で、CPU21がBank46に対しWrite RQを発行した場合のWrite動作について図4を用いて説明する。ただし図4において、第二の実施の形態のWrite RQの動作説明ではS8の代わりにS8'とし、S14の代わりにS14'とし、S15の代わりにS15'とし、S16の代わりにS16'とする。

【0072】S1でCPU21がWrite RQを転送制御装置23に発行する。以下第一の実施の形態におけるWrite RQの動作の説明のS1からS7と同様である。第二の実施の形態におけるWrite動作では、第一の実施の形態におけるS8の処理は行わずS8'の処理をする。

【0073】S8'でBus競合調停手段34による調停の結果、所属するPUのWrite RQがBusに対して発行可能な状態になると、Bus管理手段33は、該RQがWrite RQである情報を受け取り、Bus Busy管理手段491でBusのData Cycleの使用状況を確認する。BusはPipeline方式により動作する為、Write RQに対応するWrite DataはWrite RQがBus上に発行されてからLatency後にBus上に転送できるが、Latency以前に使用されていないData Cycleがある場合、Write RQがBusに発行されてから最早の使用されていないData CycleでDataをBusに転送することが可能である。この場合、Write Dataを転送するタイミング、すなわち、Data Cycleを使用するタイミングを意味する情報をBus Busy管理手段491に登録する。Bus Busy管理手段491内のレジスタ値の変化を図15に示す(第一の実施の形態におけるWrite RQ動作のS8、第二の実施の形態におけるRead RQ動作のS9参照)。Bus全体のTimingレジスタの動作は、第一の実施の形態のWrite RQ動作を同様である。また、有効Timingレジスタの動作は、第一の実施の形態と同様に時刻T+1において、Bus全体のTimingレジスタにセットされた1が所属するPUに有効であれば同じTimingで1がセットされ、有効でなければ0がセットされる。

【0074】S9からS13では、第一の実施の形態におけるWrite動作の説明S9からS13と同様である。第二の実施の形態におけるWrite動作では、第一の実施の形態におけるS14の処理は行わずS14'の処理をする。

【0075】S14'でBus管理手段49は、MM制御手段48からMMが受け取ったRQがWrite RQである情報を受け取り、Bus Busy管理手段491に該RQのWrite DataがBusを使用するData CycleのタイミングをS8'と同様に登録する。第二の実施の形態におけるWrite動作では、第一の実施の形態におけるS15の処理は行わずS15'の

処理をする。

【0076】S15'でWrite RQ発行元PU11のBus接続手段35は、Data保持手段351に保持しているDataを、S3でBus Busy管理手段491に登録してあるBusのData CycleにWrite RQのWrite DataをBus上に転送する。第二の実施の形態におけるWrite動作では、第一の実施の形態におけるS16の処理は行わずS16'の処理をする。

【0077】S16'でMM接続手段41は、Bus管理手段49からBus Busy管理手段491に登録してあるBusのData CycleにWrite RQのWrite Dataが転送される指示を受け、RQ識別手段411にて前記の指示でData CycleにあるDataをBusより受け取る。

【0078】S17からS18では、第一の実施の形態におけるWrite動作の説明と同様である。以上により、一連のWrite RQの動作を終了する。

【0079】図9は、第二の実施の形態におけるWrite RQの動作のタイムチャートである。Write RQの動作のタイムチャートにおいて、第二の実施の形態は第一の実施の形態および従来技術と同じである(第一の実施の形態のWrite RQのタイムチャートの説明参照のこと)。

【0080】次に第二の実施の形態のC Write動作を説明する。CPU21がBank46に対しRead RQを発行し、該RQがPU12内に存在するCache LineにDirtyでHitし、PU12がPU11に対しC Write RQを発行する動作について図5を用いて説明する。第二の実施の形態におけるC Write動作は、第一の実施の形態のC Write動作の説明におけるRead動作の説明とWrite動作の説明をそれぞれ第二の実施の形態のRead動作の説明とWrite動作の説明に置き換える。図5はC Write RQの動作の流れ図であり、図4のRead RQの動作の流れ図と説明、および図5のWrite RQの動作の流れ図と説明を引用する。説明には図2の名称と符合を使用する。ただし、Write RQの動作の説明を引用する場合、Write RQをC Write RQに差し替える。

【0081】すなわち、Write RQの動作時にModeによる制御と有効Timingレジスタによる制御の違いで動作結果は図10のタイムチャートとなり第一の実施形態と同様となる。

【0082】

【発明の効果】以上説明したように、本発明の第一の効果は、第一の実施の形態において、メモリに対するWrite動作が行われるRQ発行時にModeによるRQとData間の時間を可変にするのでBus上でRQとData間をLatency以下にすることが可能となる。これにより、Write RQ発行時またはC Write RQ発行時において、Dataを供給されるCPUがRQを受けてからDataを受けるまでの待ち時間をBus上でLatency以下にすることが可能となり、前記待ち時間による性能低下を押さえることが可能となる。

【0083】第二の効果は、第二の実施の形態におい

て、メモリに対するWrite動作が行われるRQ発行時に各PU、MMで同期を取るレジスタを設けたため、Bus上でRQとData間をLatency以下にすることが可能となる。これにより、Write RQ発行時またはC Write RQ発行時において、該RQによりDataを供給されるCPUがRQを受けてからDataを受けるまでの待ち時間をBus上でLatency以下にすることが可能となり、前記待ち時間による性能低下を押さえることが可能となる。

【図面の簡単な説明】

【図1】本発明の第一の実施の形態の構成を示すブロック図である。

【図2】本発明の第二の実施の形態の構成を示すブロック図である。

【図3】本発明の第一および第二の実施の形態のRead動作の流れ図である。

【図4】本発明の第一および第二の実施の形態のWrite動作の流れ図である。

【図5】本発明の第一および第二の実施の形態のC Write動作の流れ図である。

【図6】本発明の第一の実施の形態のRQ/Dataのフォーマットである。

【図7】本発明の第二の実施の形態のRQ/Dataのフォーマットである。

【図8】本発明の第一、第二の実施の形態及び従来技術のRead動作のタイムチャートである。

【図9】本発明の第一および第二の実施の形態のWrite動作のタイムチャートである。

【図10】本発明の第一および第二の実施の形態のCache間およびCacheとMM間のData転送を行うC Write動作のタイムチャートである。

【図11】本発明の第一の実施の形態のRead動作時のBus Busy管理手段におけるレジスタ値の変化の一覧を示す図である。

【図12】本発明の第一の実施の形態のWrite動作時のBus Busy管理手段におけるレジスタ値の変化の一覧を示す図である。

【図13】本発明の第一の実施の形態のWrite動作時のBus Busy管理手段におけるレジスタ値とModeの関係の一覧を示す図である。

【図14】本発明の第二の実施の形態のRead動作時のBus Busy管理手段におけるレジスタ値の変化の一覧を示す図である。

【図15】本発明の第二の実施の形態のWrite動作時のBus Busy管理手段におけるレジスタ値の変化の一覧を示す図である。

【図16】従来技術の構成図である。

【図17】従来技術のWrite動作のタイムチャートである。

【図18】従来技術のC Write動作のタイムチャートである。

## 【符号の説明】

1 1、1 2	PU
1 3	MM
1 4	Bus
1 5	制御線
2 1、2 2	CPU
2 3	転送制御装置
3 1	CPU接続手段
3 2	Bank Busy管理手段
3 2 1	Read Busy管理手段
3 2 2	Write Busy管理手段
3 2 3	Bank Busy制御手段
3 3	Mode管理手段
3 3 1	Bus Busy管理手段
3 3 2	Mode設定手段
3 3 3	Mode識別手段
3 3 4	Mode管理制御手段
3 4	Bus競合調停手段
3 5	Bus接続手段
3 5 1	Data保持手段
3 5 2	転送手段
3 5 3	RQ識別手段
3 5 4	接続制御手段
3 6	RQ転送制御手段
3 6 1	RQ解読手段

【図 6】

Address Cycle	V	Type	Mode	Bank内アドレス	Bank番号
Data Cycle	Data				

Field	内容
V	有効ビット
Type	RQの動作指示
Mode	Write動作時に有効な付加情報
Address	Bank内アドレス (アクセスを行うBankに対するBank内アドレス)
Data	転送Data

【図 8】

時刻	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Cycle	A	D	A	D	A	D	A	D	A	D	A	D	A	D	A	D	A	D	A	D	A	D
CPU	RQ																		DA			
Bus		RQ																	DA			
MM			RQ																DA			
Bankアクセス					RQ								DA									

3 6 2	RQ保持手段
3 6 3	RQ制御手段
4 1	MM接続手段
4 1 1	転送手段
4 1 2	RQ識別手段
4 1 3	接続制御手段
4 2	Mode制御手段
4 2 1	Mode識別手段
4 2 2	識別制御手段
10 4 3	Bank Busy管理手段
4 3 1	Read Busy管理手段
4 3 2	Write Busy管理手段
4 3 3	Bank Busy制御手段
4 4、4 5	Bankアクセス手段
4 4 1	Readアクセス手段
4 4 2	Writeアクセス手段
4 4 3	アクセス制御手段
4 6、4 7	Bank
4 8	MM制御手段
20 4 8 1	RQ解読手段
4 8 2	RQ待機手段
4 8 3	RQ制御手段
4 9	Bus管理手段
4 9 1	Bus Busy管理手段
4 9 2	Bus管理制御手段

【図 7】

Address Cycle	V	Type	Bank内アドレス	Bank番号
Data Cycle	Data			

注) 各Fieldは、第一の実施の形態と同様である。

【図 1 1】

	レジスタ値T	レジスタ値T+1
Bus全体のTiming	xxxxxx	1xxxxx

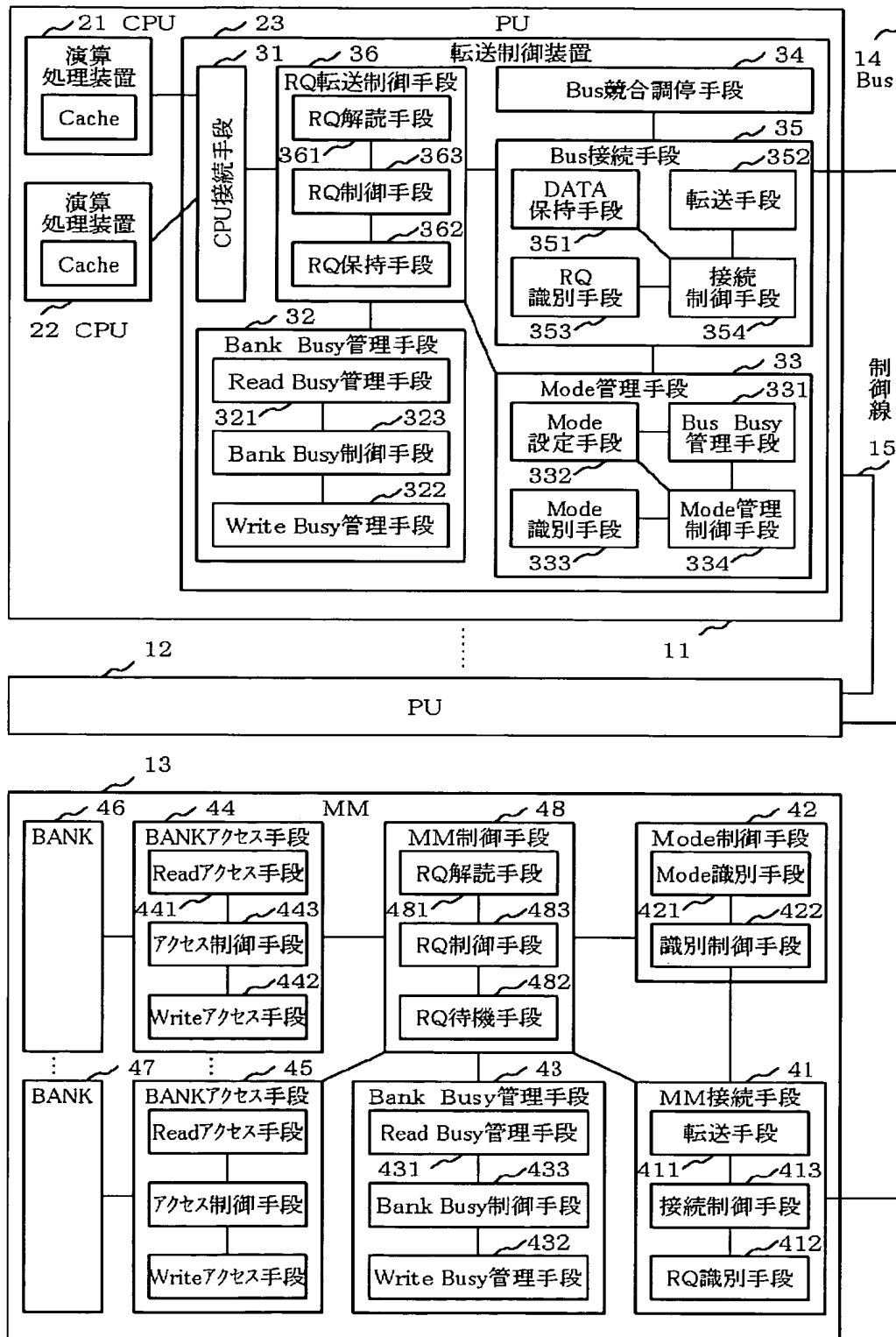
注) 新規Read RQによるBus Busyの登録があった場合xは、Don't Careを意味する

【図 1 2】

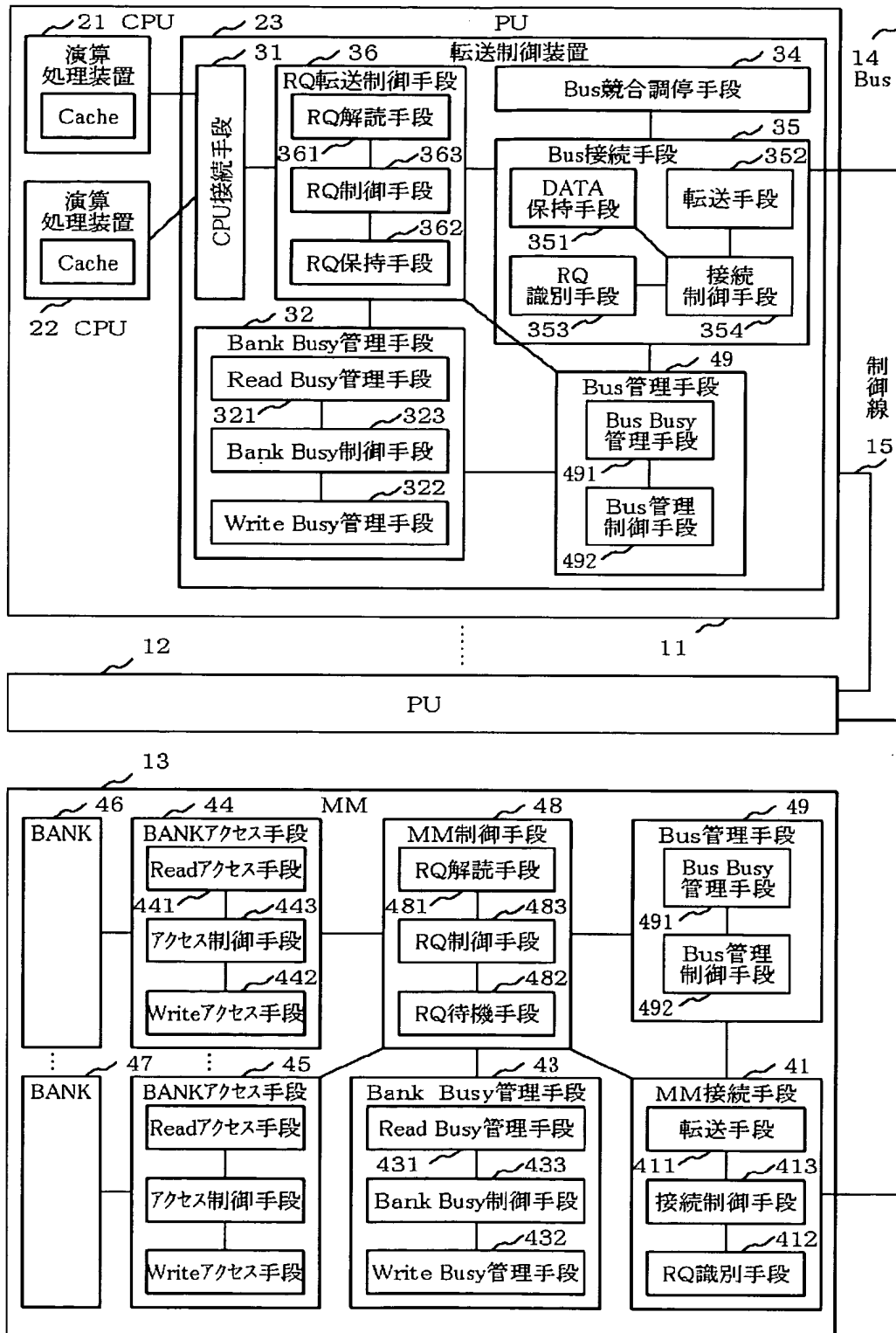
	レジスタ値T	レジスタ値T+1
Bus全体のTiming	xxxxxx0	0xxxxxx
	xxxxxx01	0xxxxxx1
	xxx011	0xxx11
	xx0111	0xx111
	x01111	0x1111
	011111	011111
	111111	111111

注) 新規Read RQによるBus Busyの登録があった場合xは、Don't Careを意味する

【図 1】

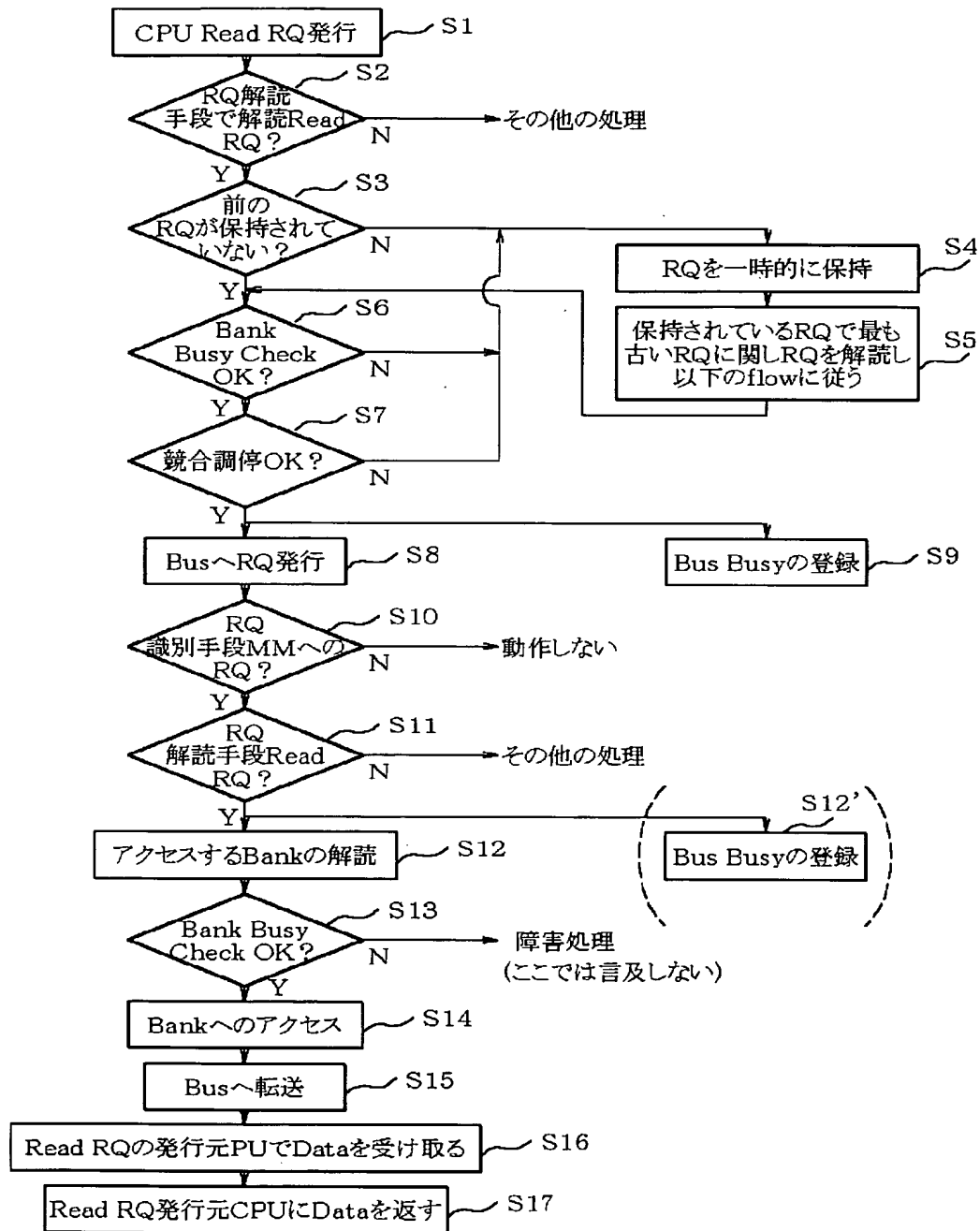


【図 2】





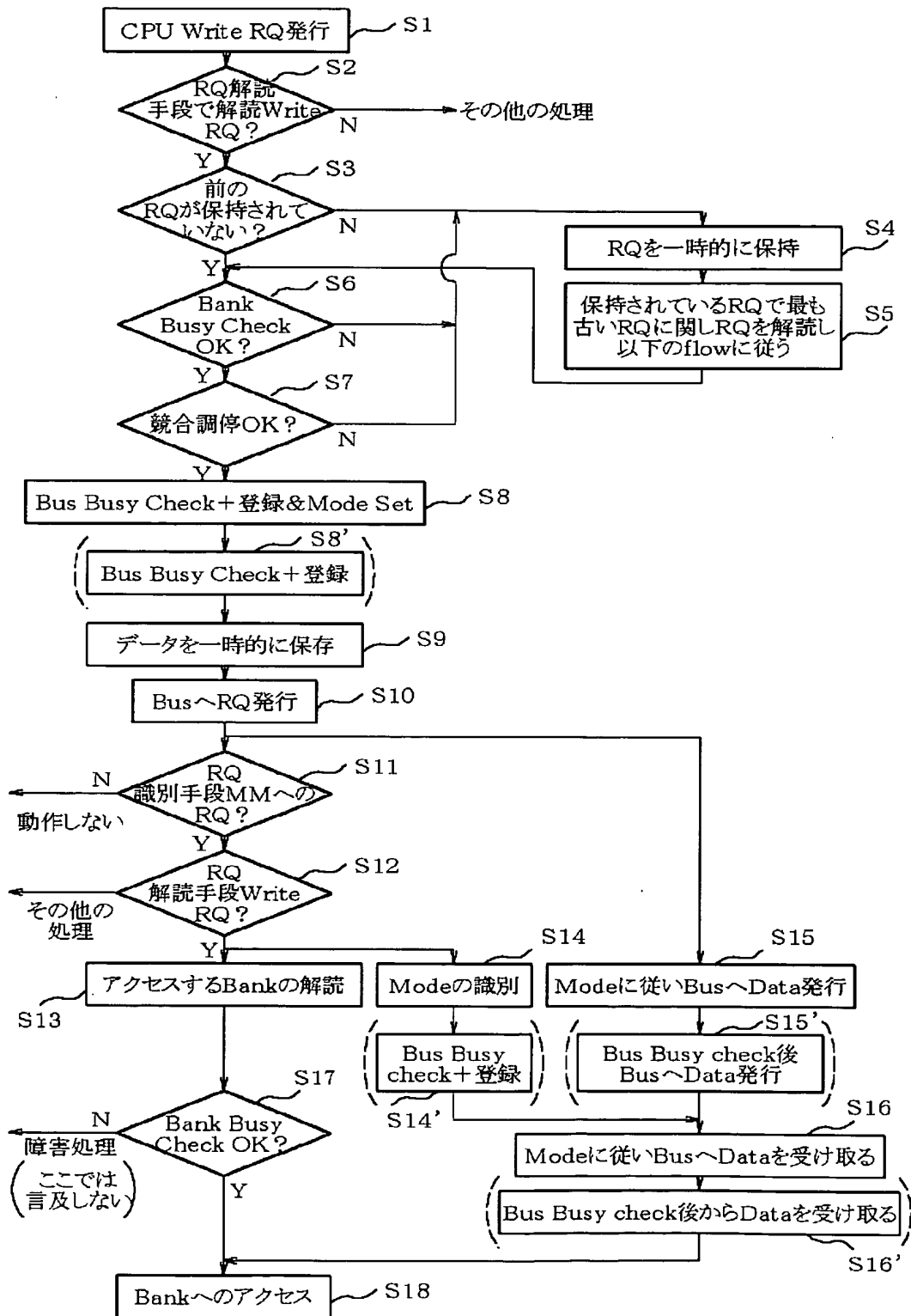
【図 3】



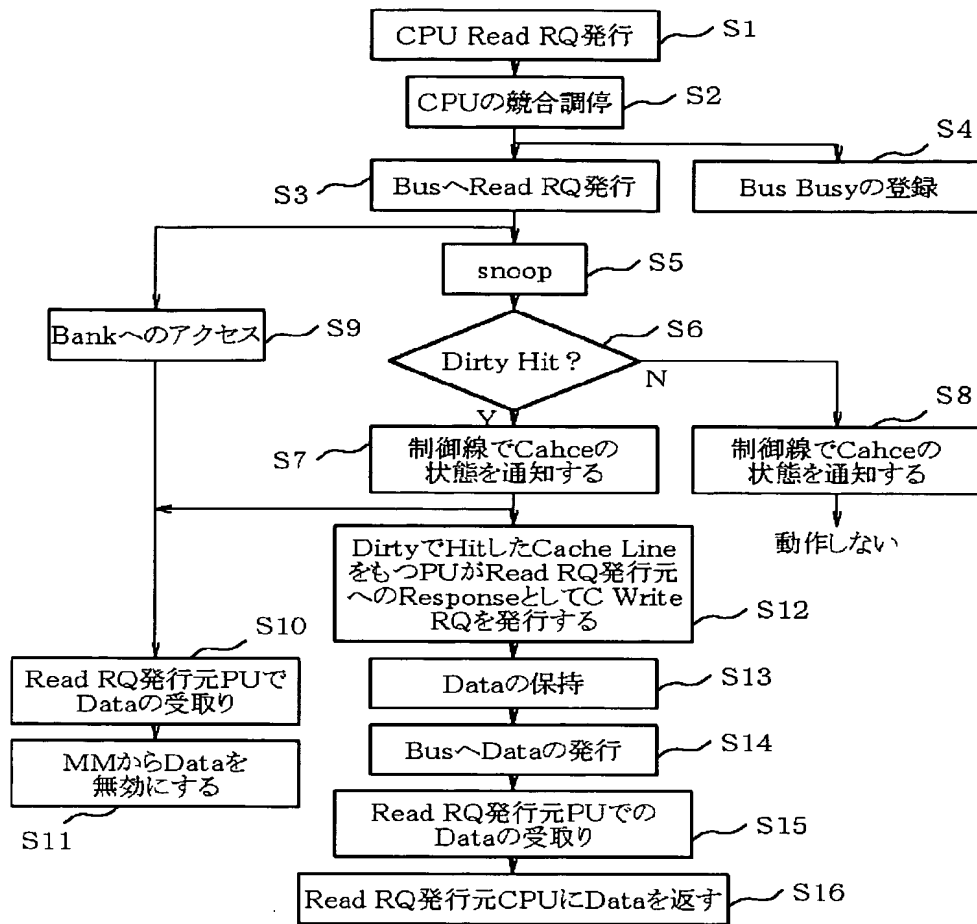
【図 1 4】

	レジスタ値T	レジスタ値T+1	備考
Bus全体のTiming	xxxxxx	1xxxxx	
	xxxxxx	1xxxxx 0xxxxx	T+1で新規Read RQによるBus Busyの登録があり、 所属するPUまたはMMに対する有効Dataがある場合は上段 所属するPUまたはMMに対する有効Dataがない場合は下段

【図 4】



【図 5】



【図 9】

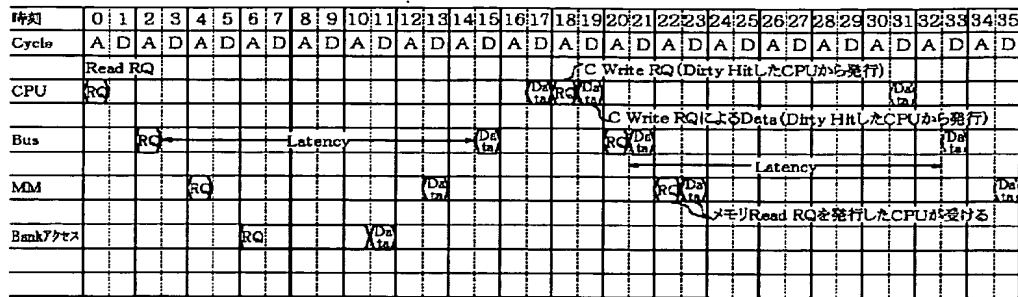
時刻	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Cycle	A	D	A	D	A	D	A	D	A	D	A	D	A	D	A	D	A	D	A	D	A	D
CPU	RQ	Da		Da		Da		Da		Da		Da		Da		Da						
Bus		RQ	Da		Da		Da		Da		Da		Da		Da		Da					
MM					RQ	Da																
Bankアクセス							RQ															

Latency  
RQの待機時間  
Write Dataのタイミング  
Bus Memory書き込み

【図 13】

	レジスタ値T	Mode
Bus全体のTiming	xxxxxx0	000
	xxxxx01	001
	xxx011	010
	xx0111	011
	x01111	100
	011111	101
	111111	110

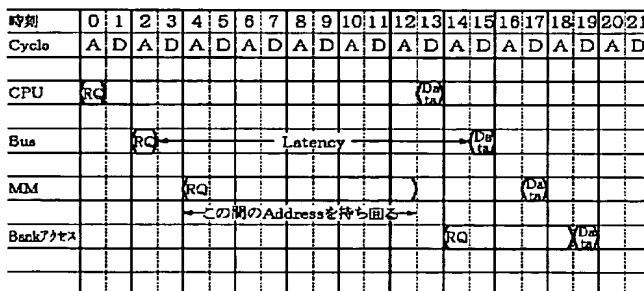
【図 10】



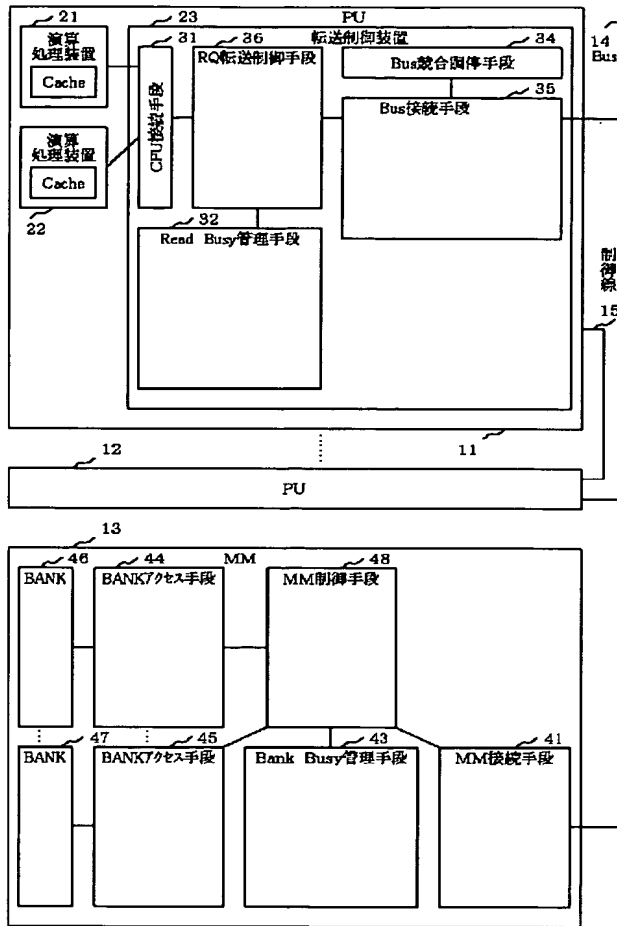
【図 15】

	レジスタ値 T	レジスタ値 T+1	備考
Bus 全体の Timing	xxxxx0	0xxxxx	Bus 全体の Timing の値は、時刻 T での最も右の 0 に対して時刻 T+1 で新規 Write RQ/C Write RQ による Bus Busy の登録 (1 が set) が行われ、Latency では有効な Data 存在しない事を示す為に最も左に 0 が Set される。
有効 Timing	xxxxx0	0xxxxx	
以下同様	xxxx01	0xxxx1	有効 Timing の値は、所属する PPU または MM に対する有効な Data が存在する場合、Bus 全体の Timing と同じ Timing で Bus Busy の登録 (1 が Set) が行われる。
	xxxx0x	0xxxx1	
	xxx011	0xxx11	
	xxx0xx	0xxxxx	
	xx0111	0xx111	
	xx0xxx	0xxxxx	
	x01111	0x1111	
	x0xxxx	0xxxxx	
	011111	011111	
	0xxxxx	01xxxx	Bus の Latency 後に有効な Data が存在する事を示す。
	111111	111111	
	xxxxxx	1xxxxx	

【図 17】



【図 16】



【図 18】

